

멋쟁이사자처럼 AI 스터디 1조

# 선형 회귀

## (Linear Regression)

김하영

박시윤

이성민

# CONTENTS

- 1 가설 수립
- 2 손실 계산
- 3 경사 하강법
- 4 다중 선형 회귀

# 1 가설 수립

시간(x)	점수(y)
1	2
2	4
3	6
4	?

1시간 공부했더니 2점,  
2시간 공부했더니 4점,  
3시간 공부했더니 6점



그렇다면, 4시간 공부하면 몇 점을 맞을 수 있을까?

모델 학습 위한 데이터는 파이토치의 텐서 형태 (torch.tensor) → 입력 : x, 출력 : y

```
x_train = torch.FloatTensor([[1], [2], [3]])
```

```
y_train = torch.FloatTensor([[2], [4], [6]])
```

**훈련 데이터셋 (training dataset)**

예측을 위해 사용하는 데이터



**테스트 데이터셋 (test dataset)**

학습 후 모델 정확성 판별하는 데이터

# 1 가설 수립

## 가설 (Hypothesis)

머신러신에서의 식. 맞는 가설이 아니라면 계속 수정해 나간다.

## 선형 회귀의 가설

학습 데이터와 가장 잘 맞는 하나의 직선을 찾는다

$$H(x) = Wx + b$$

$W$ : 가중치 (Weight) = 기울기

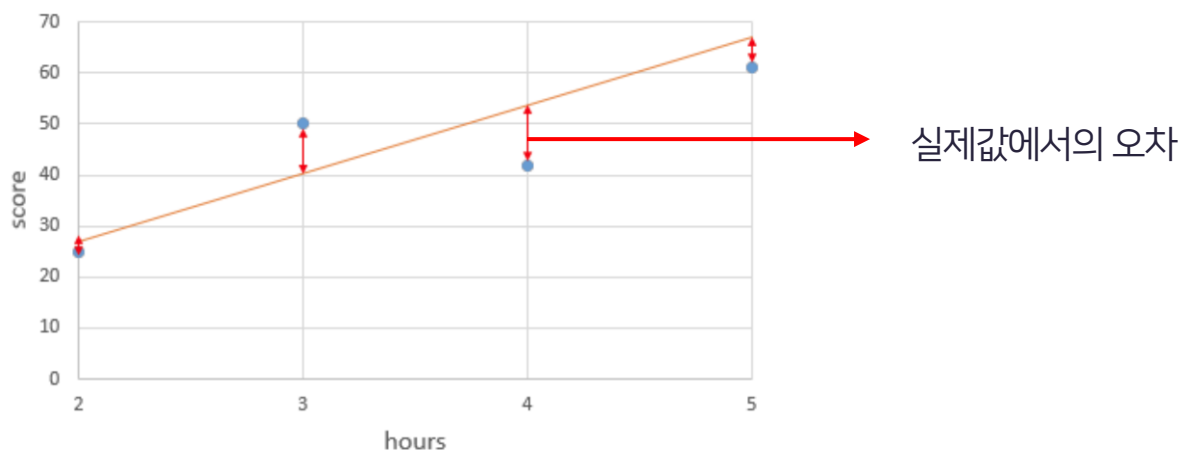
$b$ : 편향(bias) =  $y$  절편

→  $W$ 와  $b$ 의 값을 계속 수정하며 데이터를 가장 잘 표현하는 직선을 그린다

## 2 손실 계산

### 비용 함수 (cost function) = 손실 함수 (loss function)

원래의 값과 가장 오차가 작은 가설 함수. 비용함수 최소가 되게 만들어야 함.



총 오차 (total error)

$$\sum_{i=1}^n [y^{(i)} - H(x^{(i)})]^2 \quad n = \text{데이터 개수}$$

단순히 오차를 더하기만 한다면 오차값이 음수도, 양수도 될 수 있음  
→ 제곱한 뒤 더하기

### 평균 제곱 오차 (Mean Squared Error, MSE)

오차 제곱합에 대한 평균. 선형 회귀에 가장 적합한 비용 함수.

$$\text{cost}(W, b) = \frac{1}{n} \sum_{i=1}^n [y^{(i)} - H(x^{(i)})]^2$$

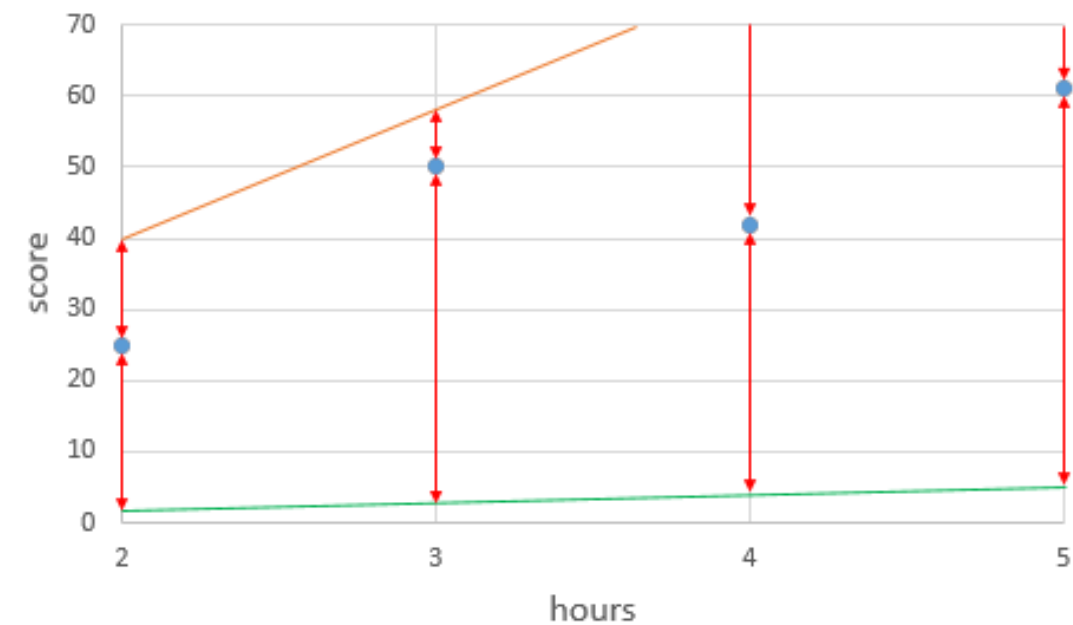
→  $\text{cost}(W, b)$  최소가 되게 만드는  $W, b$  값을 구하자

### 3 경사 하강법

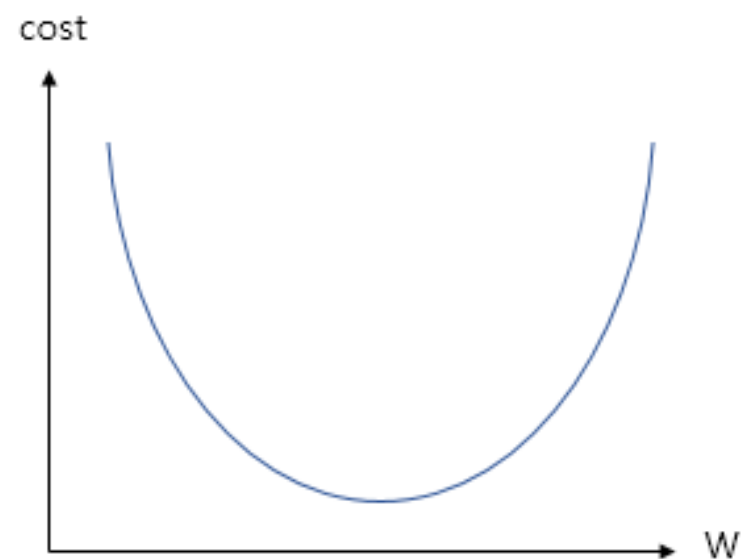
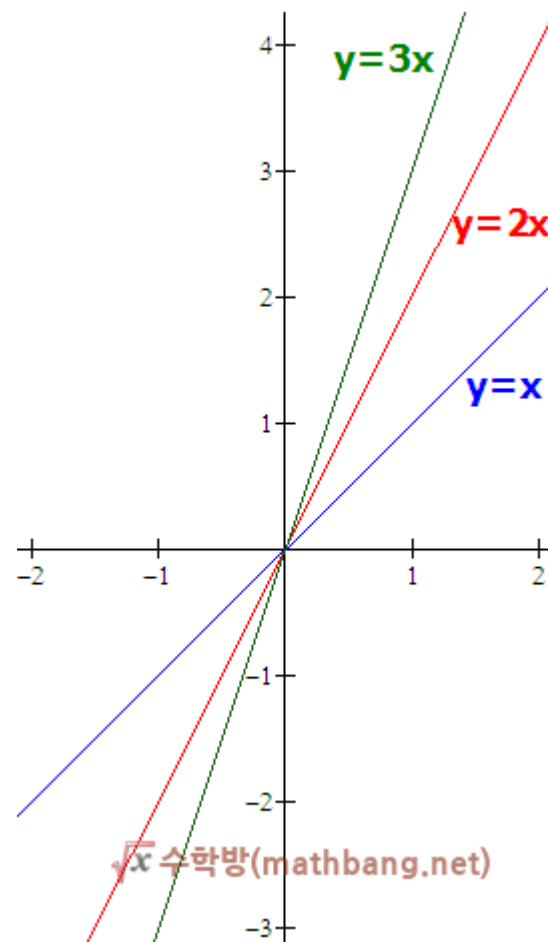
Gradient Descent Algorithm



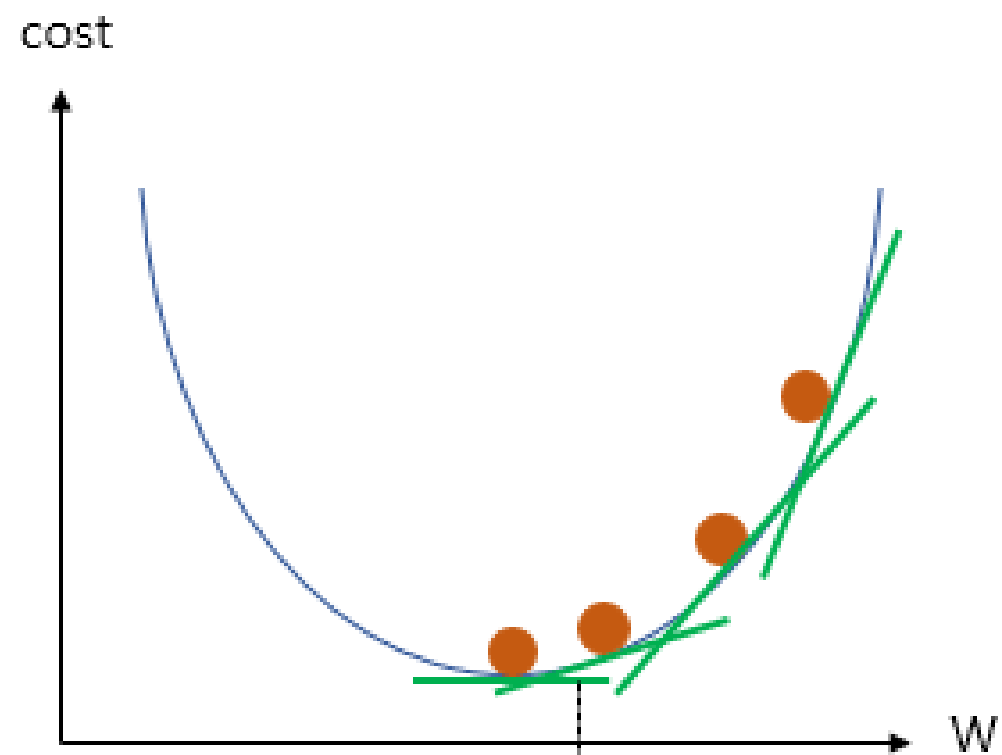
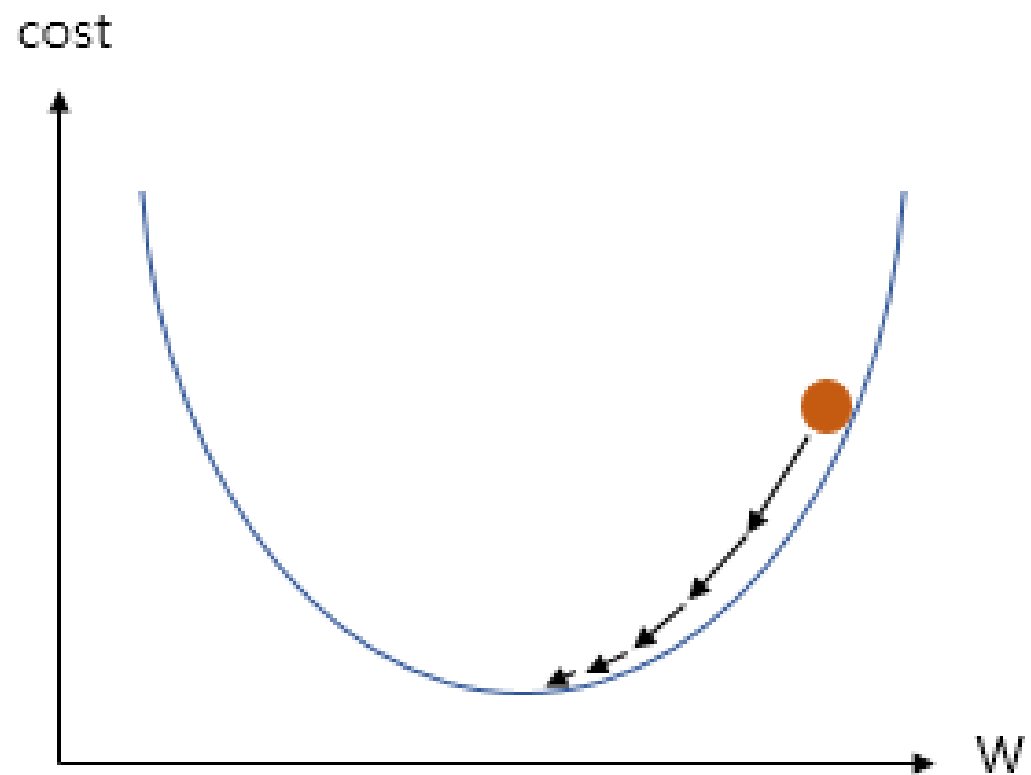
### 3 경사 하강법



$$y = Wx$$



### 3 경사 하강법



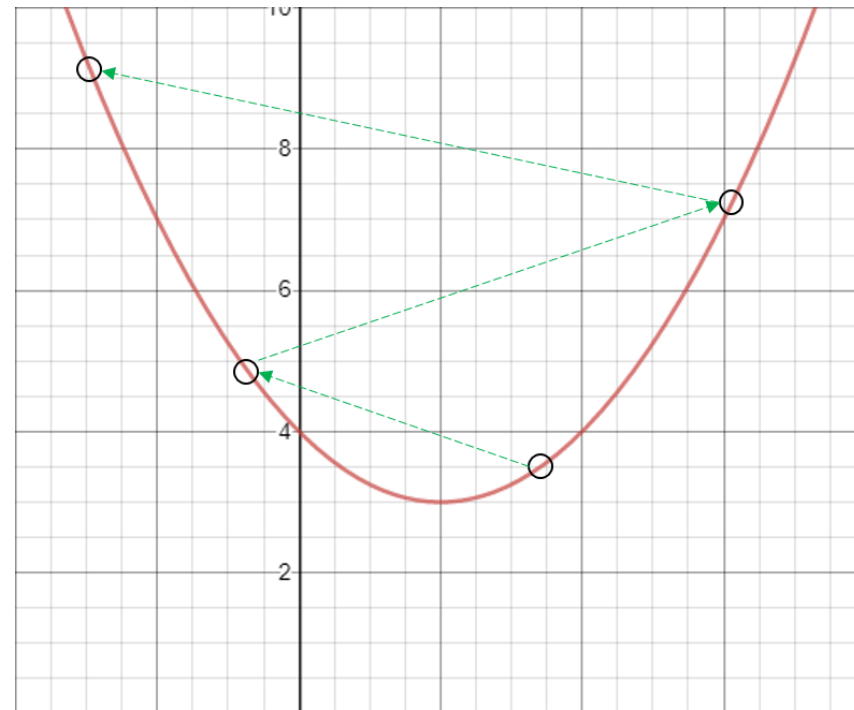
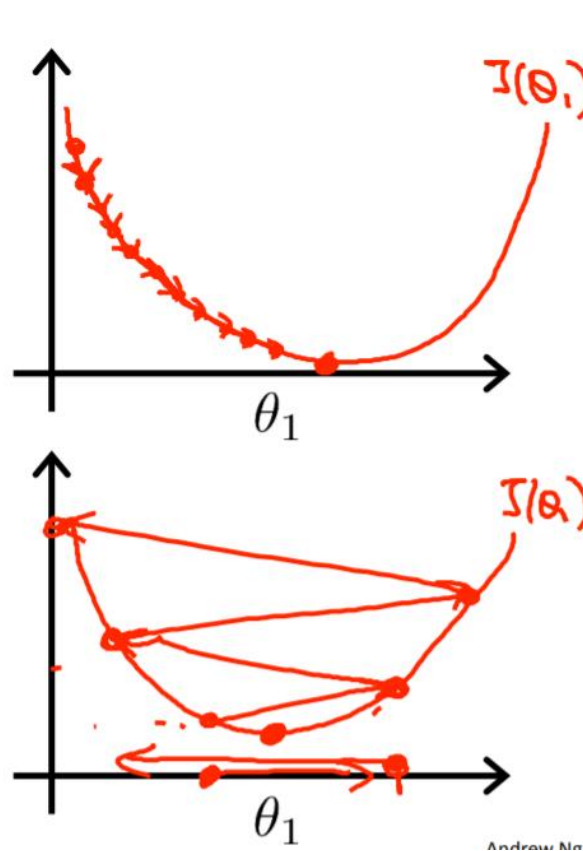
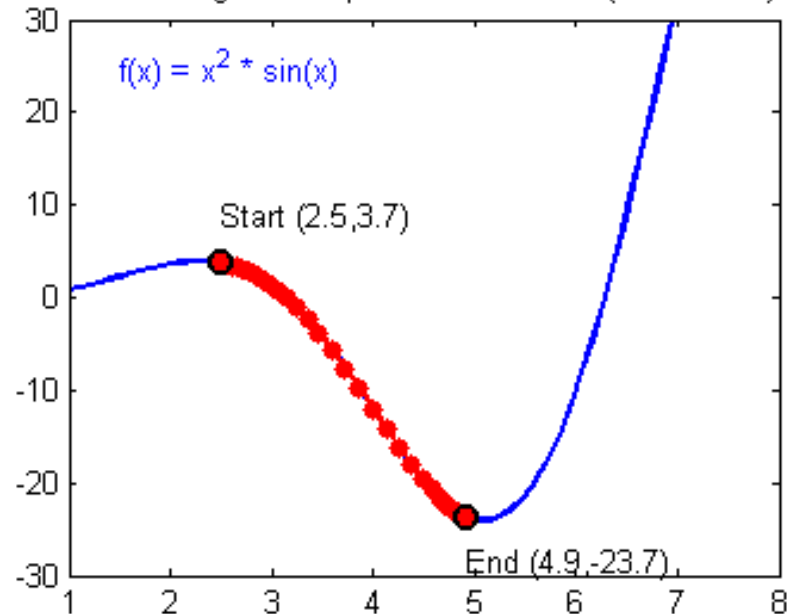
접선의 기울기 = 0



### 3 경사 하강법

$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

Descending with step coefficient 0.005 (iteration 50)



## 4 다중 선형 회귀

다중 선형 회귀  
(Multivariable Linear  
Regression)

다수의 x로부터 y를 예측하는 선형 회귀

퀴즈1(x1)	퀴즈2(x2)	퀴즈3(x3)	최종 점수(y)
73	80	75	152
93	88	93	185
89	91	80	180
96	98	100	196
73	66	70	142

독립 변수 x의 개수가 3개이므로, 다음과 같이 수식 표현 가능

$$H(x) = w_1x_1 + w_2x_2 + w_3x_3 + b$$

## 4 다중 선형 회귀

단순 선형 회귀와 달리  $x$ 가 3개이므로  $x$ 를 3개 선언

```
# 훈련 데이터
x1_train = torch.FloatTensor([[73], [93], [89], [96], [73]])
x2_train = torch.FloatTensor([[80], [88], [91], [98], [66]])
x3_train = torch.FloatTensor([[75], [93], [90], [100], [70]])
y_train = torch.FloatTensor([[152], [185], [180], [196], [142]])
```

```
# 가중치 w와 편향 b 초기화
w1 = torch.zeros(1, requires_grad = True)
w2 = torch.zeros(1, requires_grad = True)
w3 = torch.zeros(1, requires_grad = True)
b = torch.zeros(1, requires_grad = True)
```

**$x$ 의 개수가 1000개일 때도 코드 구현 시 일일이 다 선언해야 할까?**

## 4 다중 선형 회귀

이를 해결하기 위해 **행렬 곱셈 연산(벡터의 내적)**을 사용

**벡터의 내적(Dot Product)**

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 58 & \end{bmatrix}$$

$$1 \times 7 + 2 \times 9 + 3 \times 11 = 58$$

**벡터와 행렬 연산**으로 선형 회귀의 가설 표현 가능

## 4 다중 선형 회귀

### 1) 벡터 연산으로 이해하기

$$H(X) = w_1x_1 + w_2x_2 + w_3x_3$$

위 식을 두 벡터의 내적으로 표현 가능  
두 벡터를 각각  $X$ 와  $W$ 로 표현한다면 가설은 다음과 같다

$$H(X) = XW$$

$x$ 의 개수가 3개였음에도  $X$ 와  $W$ 라는 **두 개의 변수로 간단하게 표현**

## 4 다중 선형 회귀

### 2) 행렬 연산으로 이해하기

퀴즈1(x1)	퀴즈2(x2)	퀴즈3(x3)	최종 점수(y)
73	80	75	152
93	88	93	185
89	91	80	180
96	98	100	196
73	66	70	142

#### 샘플(sample)

- 전체 훈련 데이터의 개수를 셀 수 있는 1개의 단위
- 현재 샘플의 수는 총 5개

#### 특성(feature)

- 각 샘플에서  $y$ 를 결정하게 하는 각각의 독립 변수  $x$
- 현재 특성의 수는 3개

독립 변수  $x$ 의 수는 (샘플의 수)  $\times$  (특성의 수)로 총 15개!

## 4 다중 선형 회귀

독립 변수  $x$ 들과 가중치  $w$ 들을 각각 하나의 행렬과 벡터로 표현

독립 변수  $x \rightarrow$  행렬  $X$   
가중치  $w \rightarrow$  벡터  $W$

$$\begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \\ x_{41} & x_{42} & x_{43} \\ x_{51} & x_{52} & x_{53} \end{pmatrix} \times \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \begin{pmatrix} x_{11}w_1 + x_{12}w_2 + x_{13}w_3 \\ x_{21}w_1 + x_{22}w_2 + x_{23}w_3 \\ x_{31}w_1 + x_{32}w_2 + x_{33}w_3 \\ x_{41}w_1 + x_{42}w_2 + x_{43}w_3 \\ x_{51}w_1 + x_{52}w_2 + x_{53}w_3 \end{pmatrix}$$

독립 변수  $x$

가중치  $w$

위 식을 다음과 같이 간단하게 표현

$$H(X) = XW$$

## 4 다중 선형 회귀

$$H(X)=XW$$

- 위의 가설에 편향 b 추가
- 편향 b는 각 샘플에 더해짐
- 편향 벡터 B는 샘플 수만큼의 차원을 가짐

$$\begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \\ x_{41} & x_{42} & x_{43} \\ x_{51} & x_{52} & x_{53} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} + \begin{pmatrix} b \\ b \\ b \\ b \\ b \end{pmatrix} = \begin{pmatrix} x_{11}w_1 + x_{12}w_2 + x_{13}w_3 + b \\ x_{21}w_1 + x_{22}w_2 + x_{23}w_3 + b \\ x_{31}w_1 + x_{32}w_2 + x_{33}w_3 + b \\ x_{41}w_1 + x_{42}w_2 + x_{43}w_3 + b \\ x_{51}w_1 + x_{52}w_2 + x_{53}w_3 + b \end{pmatrix}$$

$$H(X) = XW + B$$

전체 훈련 데이터의 가설 연산을 3개의 변수만으로 간단하게 표현 가능



---

# Thank You

---